(2)

SAF/AQT-SR-90-005

AD-A221 592

ADVANCED RETRIEVAL ASSISTANCE FOR THE DGIS GATEWAY

Richard S. Marcus

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139

LIDS R-1958

March 1990

UNITED STATES AIR FORCE
SCIENTIFIC AND TECHNICAL INFORMATION PROGRAM
CONTRIBUTIONS TO INFORMATION SCIENCE

USAF STINFO CONTRIBUTION 90/1

SECRETARY OF THE AIR FORCE
DEPUTY FOR SCIENTIFIC AND TECHNICAL INFORMATION (SAF/AQT)
THE PENTAGON
WASHINGTON, DC 20330-1000

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
Mar 90

3. REPORT TYPE AND DATES COVERED
Final Sep 89-Mar 90

4. TITLE AND SUBTITLE
Advanced Retrieval Assistance for the DGIS Gateway.

5. FUNDING NUMBERS
C MDA903-88-C-0186

6. AUTHOR(S)
Richard S. Marcus

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139

8. PERFORMING ORGANIZATION REPORT NUMBER
LIDS R-1958

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Secretary of the Air Force
Deputy for Scientific and Technical Information
SAF/AQT, Room 4D289, The Pentagon
Washington, DC 20330-1000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
USAF-STINFO
Contribution-90/1
SAF/AQT SR-90-005

11. SUPPLEMENTARY NOTES
United States Air Force Scientific and Technical Information Program, Contributions to Information Science, USAF STINFO Contribution 90/1

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
The objective of this investigation was to further develop and test advanced retrieval assistance techniques within the experimental CONIT testbed for possible incorporation into DGIS. Significant results were achieved in three areas: (1) a partial implementation of an integrated, networkable X-Window interface, (2) implementation of the first phase of an algorithm that automatically ranks documents according to precision/relevance models, and (3) recommendations for incorporating existing advanced techniques and further design of more powerful techniques. Also developed was the design and partial implementation of an automatic search strategy narrowing selector based on user feedback or reasons for document irrelevance.

14. SUBJECT TERMS
Networking, X-Windows, Ranking, Retrieval assistance, CONIT, DGIS, Modelling

15. NUMBER OF PAGES
40

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Advanced Retrieval Assistance for the DGIS Gateway

Richard S. Marcus

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139

Laboratory for Information and Decision Systems Report LIDS R-1958
March, 1990

This is the final technical report for Task EAAA32 under Department of Defense contract MDA903-88-C-0186. The task was accomplished during the period September 15, 1989, through March 31, 1990, as subcontractor to the Control Data Corporation.

1. BACKGROUND

1.1 DGIS and CONIT

The DoD Gateway Information System (DGIS) (see, e.g., papers by Cotter [COTT86] and Kuhn [KUHN88]) has been developed by the Defense Technical Information Center (DTIC) to assist users in gaining easier access to, and make more effective use of, various computer-based information resources. For purposes of this investigation we focus on access to and use of document retrieval systems such as DIALOG, ORBIT, and ELHILL of the National Library of Medicine (NLM). The primary assistance provided by DGIS for document retrieval systems has been, until recently, limited largely to easier access; while connection and login protocols were automatically handled for the user, the user still had to contend with learning the basic search operations of each different system and database. This kind of assistance, then, tended to help only information specialists who were already expert in the use of these systems.

To provide additional assistance to DGIS users of document retrieval systems DTIC has engaged in programs to study how certain so-called front-end - or intermediary-system - techniques could be applied. In one manifestation of this effort the SearchMaestro front-end module - which provides simplified, menu-based access to retrieval systems - has been incorporated into DGIS. While SearchMaestro does make it possible for end users to perform simplified search operations, it provides little in the way of sophisticated assistance for developing effective and comprehensive search strategies of the kind a human expert searcher could be expected to perform. Attention has recently been given to how to further improve DGIS so as to incorporate more sophisticated assistance. One aspect of this effort has been the project, sponsored by DTIC in conjunction with contract MDA903-85-C0139 with Logistics Management Institute (LMI) seeking to determine the potential effectiveness in the DGIS context of techniques of the kind found in the MIT CONIT experimental retrieval assistance system.

CONIT, an acronym for "COnnector for Networked Information Transfer," includes such facilities as a common command language combined with a menu-oriented interface mode, automated procedures for converting a user's natural-language phrase topic description of his or her problem into an effective search strategy, and other mechanisms for assisting users develop effective search strategies. (For background on CONIT see papers by Marcus, e.g., [MARC81, 83, and 85].) The version of CONIT around which the above-mentioned investigation was conducted can be described as a partial implementation of an "expert" version of CONIT whose design incorporates quantified evaluations of search effectiveness as well as automated search strategy modification techniques based on *a priori* retrieval models and *a posteriori* application of user relevance inputs to determine optimized search modification procedures.

ADVANCED RETRIEVAL ASSISTANCE
CONIT CONCEPTS

PROBLEM AND SEARCH PREPARATION

- NATURAL LANGUAGE KEYWORD-STEM SEARCH STRATEGY
- INDEX TERM BROWSING
- COMMMON COMMAND LANGUAGE
- SEARCH DEFINITION AND DELIMITING
- DATABASE SELECTION
- SEARCH HISTORY AND CONSTRUCTION
- COMPREHENSIVE SEARCH OPERATIONS

SEARCH AND EXECUTION DISPLAY

- COMPONENT SEARCH RECORDING
- MULTIPLE DATABASE SEARCHING
- COMPREHENSIVE RESULTS DISPLAY
- CONNECTION FLEXIBILITY
- SOPHISTICATED CONNECTION ALGORITHM

SEARCH EVALUATION AND MODIFICATION

- RECALL ESTIMATION
- DOCUMENT RELEVANCE RANKING
- SEARCH STRATEGY MODIFICATION

SUPPORT FEATURES

- ASSISTANCE FACILITIES
- COMMAND/MENU INTERFACE MODALITIES
- USER MEMO CREATION AND DISPLAY
- COST RATES/ESTIMATION DISPLAY
- SESSION HISTORY
- KNOWLEDGE BASE MAINTENANCE
- SESSION RECONNECT
- WORKSTATION ENVIRONMENT
- WINDOWING, GRAPHICS, MOUSE/CURSOR INPUT

The aforementioned investigation included a series of experiments in which regular DGIS users ran CONIT to get answers to their current problems. A detailed analysis of these experiments was the conducted. The conclusions of this analysis (see report: Richard S. Marcus, Experimental Evaluation of CONIT in the DGIS Gateway Environment, DAITC report DAITC/TR-88/012, February, 1988) were that techniques for enhancing DGIS in a major way *could* be found in MIT's CONIT project.

1.2 Relationship of CONIT to Other Research and Development

In order to understand better the nature of the CONIT research program in general, and in this particular investigation, and to evaluate its potential for future research and development, it is useful, we feel, to review its relationships to other previous and ongoing efforts. We believe our research is unique in its *combination* of methodologies and features as well as in *particular* techniques we employ or propose. Within the realm of text-based or bibliographic databases attempts at advanced or 'intelligent' retrieval procedures may be categorized as centered on three main retrieval paradigms: (1) statistical/probabilistic/vector models; (2) deep semantic or natural language models; and (3) 'smart' Boolean models. (The traditional model with simple Boolean search using a fixed set of controlled-vocabulary [thesaurus] terms is universally recognized to be deficient.)

Vector models, as pioneered by Salton and others (see, e.g., [SALT83]), emphasize statistical correlations of word counts in documents and document collections, and lengthy problem statements taken as queries. Investigations emphasizing processing natural language or AI-like frames are exemplified by [MCCU85], [KATZ88], [RAU88], and [METZ89]. Sometimes the natural language paradigm is subsumed under the discipline of artificial intelligence; however, in fact, various aspects of AI can, and are, applied to *each* of the paradigms by some investigators in certain situations. This point of view is supported in analysis by Smith [SMIT87], Croft [CROF87a], and Belkin [BELK86]. Unfortunately, as we have attempted to point out [MARC89b], there is often too much 'hype' about 'intelligence', 'knowledge bases', 'expert systems' and the like, and too little attention to the special characteristics of the retrieval application in accounts of how AI techniques have been (might be) used to enhance retrieval capabilities.

The smart Boolean paradigm, to which we subscribe, emphasizes taking advantage of the structural and contextual information – i.e., existing 'knowledge bases' – that are typically available in a modern retrieval system along with the clarity of Boolean expression and the capability for interactive feedback associated with those systems. Of course, just as it is true that AI techniques can be applied along with any search paradigm, it is also true that most current researchers recognize that effective retrieval systems may well borrow from more than one search paradigm. Nevertheless, our own approach, we feel, while incorporating a number of features of the other paradigms, is poised to bring the smart Boolean model to a new level of utility in which it can serve as a better *basis*, in some sense, than the other models.

Three characteristics that, we believe, tend to distinguish our efforts – both within and

outside the smart Boolean paradigm – are *universality*, *interaction*, and *efficiency.* Because our early research in this field [MARC81] focussed on the integration of multiple, heterogeneous databases, we emphasized techniques that would be universal in character as well as efficient enough to be practical in operational environments. The majority of those advocating 'intelligent' assistance for searchers has promoted the idea that the appropriate knowledge base to start from is either a thesaurus in which the relationships among terms are maintained or a full-scale semantic encoding of the full database in which *all* relationships (and meanings in general) are made explicit. Some examples of this position may be found in [GAUC89], [FIDE86], [MCCU85], [POLL84], [SHOV85], and [SMIT89]. Contrarily, we believe we have identified certain fairly simple and universal procedures that, for the *document retrieval* application will prove to be as – or *more* – effective without recourse to the expense and difficulty involved in developing and maintaining thesauri or frame/slot representations or in syntactic and semantic parsing text, especially for large or multidisciplinary databases. We aver that a large percentage of term relationships can be automatically identified through word-stem overlaps among term phrases. Our techniques for automatic phrase decomposition, common word exclusion, and stemming accomplish this and permit us to relate user's natural language topic expressions to both the free text and thesaurus terms in the documents' database records.

To capture semantic relations beyond what these morphological and syntactic techniques permit we emphasize using, through *interactive, mixed-initiative* procedures, the natural knowledge base in the *mind of the human searcher* along with the knowledge base implicit in the databases themselves. Thus, the system can alert the searcher to the need for identifying semantically related terms and there is an excellent chance that either the human will be able to extract them from his own head or can be directed by the system to identify them from records in the database, especially for typical databases which have extensive text in the form of abstracts along with title words and index terms. Note that one of our newly proposed techniques – sampling from purposely broadened searches – specifically includes this prospect and obtains the required information from the user by requiring only his *recognition* from the displayed text, not any *generation* of vocabulary as such.

Note that there are other techniques and approaches often associated with AI and the other paradigms which we *do* employ. The CONIT system can provide fairly detailed explanations to the users concerning its rationale for performing most operations; the system gives explanatory information when *it* deems appropriate and the *user* can elicit particular information with the WHY and EXPLAIN commands. We have also proposed some limited use of statistics of terms and user relevance judgments. In the expert version of CONIT we have made other efforts to simulate some of the intelligent procedures of human experts such as gathering the user's problem statement. We have gone somewhat beyond what the human expert does in the way of attempting to *formalize* this statement, especially with respect to the 'Boolean Topic Representation.' On the other hand, we cannot claim to come close to simulating *all* of the human expert intermediary information specialist's talents. However, one may question whether there *is any* well-definable, unique set of characteristics for human experts in this field. In addition, as we are attempting to demonstrate, there are some *new* techniques we have been developing which may be both inherently superior to what current

human experts can now do in terms of the need for high levels of computational capabilities.

Of course, there have been a number of attempts, or proposals, to incorporate intelligence into retrieval assistance. We have considered the major themes above. Some efforts that bear particular relations to our own may be mentioned. Mischo [MISC86] described an online catalog in which field level was automatically adjusted by the system to obtain 'good' results. Koll has indicated [FOX89] that his SIRE system will automatically adjust the coordination level for better results; in general, however, SIRE uses statistics for its 'smarts.' Several investigators have outlined, at least, rather general systems which could potentially involve a wide variety of techniques in an intelligent manner. A number of these efforts, however, put their greatest emphasis on thesaurus-based techniques (e.g., [GAUC89], [SMIT89], and [BRAJ85]). Others put their emphasis on using statistical retrieval techniques (e.g., [FOX86], [CROF87]).

2. OBJECTIVE OF THIS INVESTIGATION

The objective of this investigation was to further develop and test advanced retrieval assistance techniques within the framework of the experimental CONIT testbed in preparation for their possible incorporation into DGIS. Development was to be done so as to provide, as much as possible, software that could easily be incorporated into DGIS/DTIC environments. The work on this task was to augment and complement work already ongoing on the CONIT project so as to enable faster and more comprehensive implementation of the techniques as well as enabling special attention to developments that are compatible with the DGIS environment. In addition to the development and testing efforts themselves a final report (this one) was to be written in which an analysis of the potential efficacy of ALL advanced CONIT techniques, whether or not a full implementation and testing for each was possible in the course of this task per se, would be given along with a discussion of what further steps would be advisable in order to incorporate the techniques into DGIS.

3. ENHANCEMENT TECHNIQUES CONSIDERED

The techniques potentially providing enhanced DGIS capabilities which were investigated are those currently in the CONIT system or designed as part of an advanced "expert" version of CONIT. Enhancements may be characterized as being in 4 areas: (1) problem and search preparation; (2) search execution and display; (3) search evaluation and modification; and (4) support and general. As discussed in the 1988 Marcus report [MARC88a], DGIS by itself offers very little in the way of assistance in *any* of these areas. However, the SearchMaestro (SM) module does provide some assistance; comparisons will be made, then, primarily with this mode of DGIS utilization.

3.1. Problem and Search Preparation

3.1.1 Natural language keyword-stem search strategy

As analysis of the CONIT DGIS experiments demonstrated in support of much previous analysis, the CONIT methodology of creating a search strategy by performing a Boolean intersection of all-(subject)-fields truncation searches on the stems of keywords automatically extracted from users' natural language topic phrases is a very powerful tool for assisting searchers, especially inexperienced searchers in a multiple, heterogeneous database environment. In contrast, SM does very little to aid users develop effective search strategies, essentially only providing a simple example or two.

3.1.2 Index Term Browsing

CONIT permits a searcher to browse through the index terms from the curr- t database. It also allows users to select terms from the inde: by tag *numbers* (instead of having to type out the full term) even in cases where the retrieval system itself does not provide those numbers. These features are not available in SM.

3.1.3 Common Command Language (CCL)

A CCL permits users to express their requests in one common form and have the system perform the appropriate translations to the currently connected retrieval system. Efforts in developing a CCL have been undertaken by both the DTIC SPO AI group and Telebase for its EasyNet family (of which SM is one member). The CCL in CONIT has additional features beyond that which has been achieved in either of these two efforts.

3.1.4 Search Definition and Delimiting

CONIT assists the user to prepare a formalized problem statement including a conceptu-

alization based on a Boolean Topic Representation and a set of known and desired problem initial conditions and limits for such aspects as recall, costs (money and time), and document types. Besides helping the searcher be more clear about his needs, this formulation can be utilized later in retrieval evaluation and search modification, as indicated below.

3.1.5 Database Selection

CONIT maintains a directory of all databases accessible on any of the retrieval systems and assists users in finding databases of relevance by leading them through a hierarchically arranged listing of the databases. Brief information about each database is immediately available in the listing and more detailed information may be requested from the retrieval systems themselves. Databases, either individually or in sets, may be selected by the user through indication of category or subcategory, full classification term, or character string or word at beginning of, or included in, CONIT full name or classification term or in alternate database names.

DTIC has developed a plan for the online utilization of a database directory in DGIS. SM has a rudimentary form of database selection assistance in which the user selects answers to a series of menu options leading to the selection of a *single* database. Besides the above-described full directory browsing scheme, CONIT has experimented with two techniques which provide a listing of databases *ranked* according to their likely relevance to the topic at hand. One such technique employs, as does SM, a series of menu formatted questions; ranking is accomplished through the application of a MYCIN-like formula operating on the user's answers. A second technique involves the searching of a multidisciplinary database (e.g., NTIS) by the keyword-stem techniques outlined in Section 3.1.1 and the cumulative transformation of classification terms found in the retrieved documents through a matrix whose rows are associated with classification terms, whose columns with databases, and elements with estimated relevance of each database to the topic expressed by the classification term.

3.1.6 Search History and Construction

Whereas SM primarily considers each search a separate monolithic entity to be discarded after its retrieval results have been initially reviewed, CONIT keeps track of all searches and search components and permits these to be operated on at any time during the session. Operations include (1) display of any search results and documents (whether or not previously seen and whether or not still current in the retrieval system – in the latter case regeneration of the "lost" search is automatically performed); (2) construction of new search strategies from old (note this can take place in CONIT itself while NOT connected to the retrieval system – thus saving costs); and (3) search evaluations and running in (additional) multiple databases (see below for more details). In addition to saving search history for a given session, CONIT has a search catalog facility which enables search strategies to be saved for future session, even allowing sharing with other individuals or groups of searchers.

3.1.7 Comprehensive Search Operations

CONIT permits a full range of proximity specifications in searching on multiple words and terms (e.g., number of intervening words or inclusion within same [unspecified] field) whereas SM is limited to the simple AND or strict adjacency specifications (i.e., just the ends of the proximity spectrum). As indicated below, plans to have CONIT be able to distinguish and specify any combination of subject-oriented fields for searching have been culminated in this project. SM is limited to a few variations of field searching.

3.2. Search Execution and Display

3.2.1 Component Search Recording

CONIT takes a compound (multi-word) search and breaks it up into its components which are then individually searched and may be separately reviewed or combined to form new compound searches (similar to the Dialog search steps facility).

3.2.2 Multiple database searching

CONIT permits a user to specify an ordered list of databases. Users may then request a search on one or all of the databases with a single request (even if the databases are on different systems). Note that the keyword-stem search methodology (cf. Section 3.1.1) makes it feasible to get good results with the same search strategy across multiple, heterogeneous databases. The EasyNet "SCAN" feature is a start toward a true multiple database searching facility; SCAN is limited to searching a few preselected sets of databases found in *one* system: Dialog. Note that in CONIT's "virtual system" approach users are not limited to framing their search for one (particular) system at a time.

3.2.3 Comprehensive Results Display

CONIT in its CCL, or through menus, allows searchers to select from four retrieval set output formats, two output presentations (online or offline), and any range of document records. SM is more limited in its flexibility. Also, it may be note that CONIT's CCL may provide at least a methodological arguing point in considering the extension of DTIC/SPO's CCL: whether to adhere to the proposed NISO CCL standard in allowing the user to present arguments to the command in *any* order and whether to try to make the command as much as possible common (or universal) across systems (as opposed to making it system dependent). Also, the question of how to handle the interface between command and menu modes arises.

3.2.4 Connection Flexibility

CONIT permits user control of whether the remote retrieval system stays connected or not. One reason to want to stay connected is to avoid having to regenerate a search. (Of course, one must tradeoff quicker response time against a possible additional cost for longer

retrieval system connect time.) CONIT does have a timer which warns of excessive connection time without interaction; such warnings could be replaced by automatic disconnection.

3.2.5 Sophisticated Connection Algorithm

CONIT chooses among many alternate connection paths for a given database (e.g., X25 or dialout, Telenet or Tymnet, which system) not only on the basis of which system is *supposed* to have that database, but also in consideration of which system is currently connected (avoid switching if not necessary) and which systems are scheduled to be available at the current time as well as making a dynamic choice depending on which paths have proven to be more reliable in the recent past (minutes and hours).

3.3. Search Evaluation and Modification

The techniques in this section include some that have been fully implemented and tested and some that have been designed but not yet implemented.

3.3.1 Recall Estimation

Three mechanisms for recall estimation have been implemented: one based on an *a priori* comprehensiveness model (how many search terms and databases searched for each conceptual factor) and two based on *a posteriori* relevance judgments as compared with, respectively, known and estimated numbers of relevant documents from *a priori* problem specifications (see 3.1.4). A fourth, and potentially most definitive, estimation technique has been designed; it is based on sampled relevance judgments from purposely broadened searches.

3.3.2 Document Relevance Ranking

Designs have been made (and implemented in this project) to enable the system to automatically rank documents on their expected relevance to the search topic. This technique is based on a model of how variations in search strategy relate to a metric of degree of association of a topic in relevance terms. Again, the problem conceptualization (cf. Section 3.1.4) is central to the implementation of this technique.

3.3.3 Search Strategy Modification

In the first stage of the expert version of CONIT, which has already been implemented, the user can be led through a series of menus to modify his search strategy to achieve designated goals (e.g., raise recall or precision). These menus include the selection from system-maintained lists of pertinent techniques for strategy modification. The design for the full-fledged expert CONIT indicates how user relevance judgments on individual documents can lead to the automation of strategy modification technique selection, thus avoiding requiring the user to become involved with details of the techniques. Note that implementation of these enhancements has been assisted by the mechanism, already initiated, of providing

modification *operators* that. when applied to existing searches, generate the searches that relate to the original searches in specified ways (e.g., change from a particular close proximity specification to an indicated looser one).

3.4. Support and General

Listed below, without elaboration, are a number of items related to user assistance and support or to general assistance system development issues.

3.4.1 General HELP, EXPLAIN, and ASSISTANCE facilities.

3.4.2 Integration of command and menu interface modalities.

3.4.3 Recording user comments and full or partial recording and printout of user's session. User memo creation and display.

3.4.4 Database and system cost rate display. Dynamic cost estimation in session. User and group cost accounting. Individual and group password control.

3.4.5 UNIX Shell command history and editing.

3.4.6 Production rule maintenance of protocol knowledge base.

3.4.7 Reconnect to dropped (or purposely *suspended*) *process*.

3.4.8 Workstation environment (extended implementation in this project).

3.4.9 Windowing, bit-mapped graphics, mouse/cursor input (initial implementation in this project).

4. WORK PLAN

The general plan was to speed up and extend the current MIT CONIT project investigations with emphasis on those aspects that are particularly appropriate to DTIC needs. In that respect, software and system developments were to be molded as far as possible so as to prepare the way for efficient incorporation of CONIT techniques into the DGIS environment. The existing CONIT software and computer systems environment were to be used as the framework and testbed for the additional developments.

While each of the techniques discussed in Section 3 above were to be considered in the analysis, a particular subset were to be emphasized and incorporated into the extended CONIT system developed under this task. The extended CONIT system was intended to serve as a demonstration and experimentation vehicle. It was intended to be usable to demonstrate the extended, as well as the existing, CONIT capabilities and for experimental testing as desired by the DTIC/DGIS community or others.

The new capabilities that were to be included in the extended CONIT system are enumerated below:

4.1 Extended Access and Windowing

Direct and/or local area network (LAN) access were to be available from existing workstations at the MIT LIDS lab including two SUN and four MicroVAX workstations. Remote access (e.g., for anyone who can now access DGIS) were to be made available through TELNET/INTERNET, RLOGIN/UUCP, or some combination of TELNET, RLOGIN, and modem dialout (e.g., through DGIS dial or any standard modem dial).

The CONIT program would, of course, be runnable as a separate process in its own window in those systems that support windows. Investigation was to be undertaken to determine how the CONIT program itself can best be structured so as to enable windowing *within* CONIT. A related aspect was to demonstrate how UNIX shells and (X) windows can be called *from* CONIT.

4.2 Simplified and More Effective Assistance Modes

Following conclusions reached in the 1988 Marcus report [MARC88a], efforts were to be directed toward providing a simplified mode of assistance in which the amount of explanatory text presented to the learning user is much reduced. This was to be achieved by simplifying and streamlining existing explanations and by withholding much of the explanatory text currently forced on the user. For this mode it will be assumed that most users do not want to learn the rationale and details of searching and that they either will be satisfied with some degradation of performance this entails or will seek the additional explanatory information which will still be available as a (not highly promoted) option. Another simplification tech-

nique would be to increase the depth of hierarchy of the menus, thus allowing reduced menu size.

Effort was also to be expended to fix bugs in the existing code, add elements of the old CONIT PL1 code that have not yet been converted to C – especially where ease of use is thereby enhanced, and otherwise "smoothing" interface difficulties. The resulting *intermediary assistance system* – even without taking into account the major new functional capabilities discussed below – will then be suitable as a testbed for demonstrations and full-scale experimentation as was concluded in the 1988 Marcus report [MARC88a] to be an important next step in development of DGIS.

4.3 Automatic Search Modification through Relevance Feedback

The user was to be prompted to give relevance/utility judgements on individual documents and, more particularly, when not relevant, to choose from a list of reasons for irrelevance/inutility. These relevance judgements and explanations were then to be used to select appropriate search strategy modification techniques. The search modifications and revised search execution would then be automatically performed, unless the user at his discretion intervenes to the contrary.

4.4 Automatic Relevance Ranking

Automatic document relevance ranking within a retrieval set was to be achieved by automatically searching the same topic using a series of increasingly more precise search strategy formulations. Degrees of precision were to be specified according to a model developed on the CONIT project which identifies degrees of matching preciseness along three dimensions: field matched (e.g., abstract words, subject index terms, and title words), exactness of word matching (e.g., truncated stem versus exact word), and inter-word proximity (e.g., record-as-a-whole [Boolean AND], same sentence, and adjacency). The full implementation of this scheme would require the addition of differentially searching among the several topic oriented fields.

4.5 Other Capability Extensions

Each of the above extended capabilities (4.1 - 4.4) were definitely be implemented and incorporated into CONIT in a demonstrable form. To the extent that time and resources permit consideration was to be given to incorporating additional extensions as outlined in Section 3. High priority was to be given to windowing and graphic extensions and to automated recall evaluation through sampled relevance judgements from purposely broadened searches.

5. WORK ACCOMPLISHED

The four specific tasks described in Sections 4.1-4.4 were each accomplished at least to the level promised. In addition, three areas that were pursued beyond the nominal levels described in the task objectives are: (1) X window design and prototype development, (2) automatic relevance ranking of documents, and (3) plans for further testing and development of advanced retrieval assistance functionality for the DGIS environment. The specific task accomplishments are described in the remainder of Section 5 and the planning recommendations in Section 6.

5.1 Enhanced Interfaces through Windowing

5.1.1 Overview and Desiderata

Over the last decade tremendous improvements have been made in providing the computer user with user-friendly, intuitive Graphical User Interfaces (GUI). In the early Eighties, Apple Computer, inspired by the research work performed in the Xerox Palo Alto Research Center, introduced the Macintosh User Interface which has since then become a standard by which to judge the user-friendliness of personal computers and workstations. In the personal computer arena, Microsoft Window and Presentation Manager are targeted towards users of MS-DOS and OS/2 respectively. In the workstation arena, initially each vendor had its own proprietary kernel-based windowing system; this heterogeneity made it difficult for window-based applications to be ported from one platform to another. However, Project Athena at M.I.T. has created the X window system, which has become the de-facto industry standard. In the near future, one may anticipate that all workstation vendors will be shipping the X window system along with their products. Using the X window system, GUI similar to the Macintosh User Interface can be developed and ported easily in multiple hardware platform.

CONIT was originally developed in the Multics mainframe PL/1 environment. Technology breakthroughs in VLSI and RISC made it possible for us to port CONIT from a mainframe to a workstation in the late Eighties. However, our initial workstation version of CONIT has inherited the mainframe characteristics in computer- user interaction. Typically the user interacts with CONIT by typing in commands or selecting from menu choices through the workstation keyboard. The CONIT staff (see, e.g., theses by William Lee [LEE85] and Hing Fai Louis Chong [CHON88]), however, have identified several areas in which GUI techniques could greatly increase efficiency and effectiveness in searching. Some improvements are quite obvious and straight forward. For example, CONIT commands can be classified into groups (e.g. search construction, search execution, search results evaluation, file selection, etc.) and be embedded in a pull-down menu structure. This would allow an occasional CONIT user, who usually does not remember the spelling of CONIT commands or their abbreviations, to browse through the pull-down menu structure and quickly identify the correct command to execute.

Another area of improvement which we have identified, however, requires more research to verify its applicability. A search session can be thought as a two-way information exchange between the user and the computer intermediary in which the computer tries to understand the user's search problem and construct the optimal strategy to perform the search and present the search results effectively and efficiently. During a search session, CONIT presents different types of information to the user such as actual documents retrieved, listing of search strategies and their results, explanations of CONIT commands, ASSIST tutorial information and menu choices. Standard CONIT presents this information in a linear form in which any new information displayed, which is usually triggered by the user entering a CONIT command or picking an ASSIST menu choice, always replaces the existing information displayed on the screen. Such simple paging and scrolling is less than optimum when the user wants to revisit certain information, such as the explanation of a CONIT command or the current search strategy. To overcome this limitation, which was inherited from the older mainframe user-interface paradigm, we addresseed the issue of using the windowing capabilities of the workstation environment by assigning different types of information output to separate windows.

5.1.2 Preliminary Window Design and Implementation

As an initial attempt to investigate the potential utility of incorporating graphic user interface techniques within an advanced retrieval assistant intermediary environment, we have implemented a preliminary version of CONIT having a windowed interface as an *additional* mode of user-computer interaction. In this preliminary implementation the user is presented initially with a rectangularly oriented overall CONIT window area that is divided into four horizontal windows whose dimensions can be adjusted by the user: (1) a menu bar, (2) a command input window, (3) the main output display window, and (4) a special explanation display window.

The top window is a menu bar having, currently, boxes labeled by four categories of user requests: HELP, FIND [search construction], SHOW [displaying documents, strategies, etc.], and MISCellaneous [connecting, running, disconnecting, and quitting]. When the user points and clicks with the mouse on one of these boxes a pull down menu of of particular operations in that category is displayed in which the user can make a further selection by pointing to the desired option and releasing the mouse button in the (now) traditional fashion. If the selected operation requires additional user input (e.g., search arguments), a popup dial box appears for that purpose.

When the user completes his input the corresponding CONIT command is shown in the main output window after which the results of the execution of that command are displayed (also in the main window unless the command was one requesting some explanation in which case the explanation is shown in the explanation window). The user also has the option of entering commands directly by typing command text from the keyboard into the command window. We believe that a well-designed pull-down menu structure would make CONIT commands more accessible by the user during a search session. It would help overcome the problem of infrequent user not being able to remember the spelling or abbreviations of

CONIT commands.

Preliminary design has been developed by which to extend CONIT to display different types of output in different windows. Our hypothesis is that multiple windows would help organize CONIT output in a fashion such that information previously displayed can be easily accessed again during a search session. As mentioned above, we have already added an explanation window so that online explanations are displayed in this special window rather than the main window. The user may scroll back and forth in the explanation window to access any explanation during the search session. In addition, we have prepared the general design for separate windows into which ASSIST messages may be displayed. The user would be able to select ASSIST menu choices by using the mouse to click on a popup menu item in addition to the current method of entering the choice number using the keyboard.

We view the point-and-click input method as an *alternate* means of collecting input from user. Some users, especially the more experienced ones, can be expected to find typing the commands or the abbreviations at the keyboard more efficient. Therefore, CONIT should always accept keyboard input from the user. We have implemented our windowing extensions in such a way that it did not affect our existing CONIT code and the user may instruct CONIT to operate in either the window or non-window mode at start up time. In fact, if the user wants to use CONIT in the window mode but CONIT discovers that the terminal capabilities are not sufficient, CONIT will gracefully fall back to the non-window mode and be still usable.

5.1.3 Technical Problems Encountered

The first problem we faced was how to allow one CONIT version to support two different user interface paradigms without making the code messy and difficult to maintain. We emphasize the importance of one CONIT version because it is costly to maintain two parallel CONIT versions merely for supporting different user interface mechanisms. To support windowing extensions, CONIT can no longer operating under the simple paradigm of displaying output to the user using the standard UNIX/C output library functions such as printf or putchar, and reading user keyboard input with the standard library input functions such as scanf or getchar. In addition to handling keyboard input, CONIT has to manage output to multiple windows and be able to buffer and process input from each window asynchronously. Another problem is that, similar to most C programs, the existing CONIT code has used the standard output libraries functions (e.g. printf and putchar) freely and, therefore, many of these function calls currently reside in almost every single CONIT module. The task of going to every such instance to change it to call a new windowing output function would be very tedious and time consuming.

5.1.4 The X Window Solution

Fortunately, we have found rescue in the object oriented paradigm of the X window system and the UNIX capabilities of supporting multi-tasking, and inter-process communications. Without the X window system, or a similar windowing system, a programmer would have

to spend much time in struggling with the low-level detail of turning pixels on the screen on and off as well as keeping track of the activities of the mouse. The X window system removes the pain of performing such tedious tasks by providing layers of abstractions to the programmer. In particular, the X window system provide an event abstraction so that user inputs at one or more windows are automatically queued. The user program may then extract the event when it is ready to process it. The X window system also support higher level abstraction in a object-oriented manner to relieve the programs from the tremendous task of maintaining data structure for multiple windows and window elements. The X toolkit intrinsics provide an object-oriented environment for widgets creations. Widgets represent window objects and they can be combined to form the user-interface suitable for different windowing applications. In our current implementation, we have used the Athena widget set available in the X window system Release 4. We, however, plan to use the Motif Widget set from Open Software Foundation in further software development; the Motif set provides more widgets than the Athena Set and it is getting wider industry acceptance as a standard.

5.1.5 Window Manager Process Design

Our next question was: should we go into CONIT and change all the printf, putchar, scanf and getchar function calls into X window function calls? We decided not to do that. Instead we put all the new code which handle interaction with the X window system in a single module which we named as window_mgr. We run window_mgr as a separate process under UNIX in parallel with existing CONIT main process. The window_mgr process communicates with the main CONIT process using interprocess communications mechanisms over the Berkeley socket interface. In this way, all the additional knowledge concerning the usage of X window systems is centralized in the window_mgr module and does not spread widely in the main CONIT code in the manner that the printf and putchar functions did. To allow main CONIT to communicate with the window_mgr process, we added a module called window_lib.c. Window_lib.c contains a window_connect function, which is called upon by CONIT during start up initialization to either start a window_mgr background process running locally on the same machine or attempt to connect to a window_mgr, which may be running physically on a different machine across the network. If the window_connect function fails to create or connect to a window_mgr process, CONIT will gracefully allow the user to operate in the non-window mode.

The window_mgr acts as a command preprocessor. Any user mouse input and textual input to window dialog boxes is transformed into a character string equivalent to traditional CONIT command line input. The character string is then sent from the window_mgr process to the main CONIT process over the socket interface. In the main CONIT side, we have modified the places which read user input from standard input to monitor the socket connecting to the window_mgr for user input. Fortunately, previous attempts at object-oriented modularity in CONIT left only a few places that had to be modified as far as collecting user input is concerned.

Unfortunately, we were not so lucky when outputing to windows was concerned. CONIT has enumerable calls to printf and putchar which spread around in every single corner of

CONIT. We were able to circumvent the need to replace these myriad calls individually with window specific output calls by virtue of our design of running window_mgr as a separate process. Since the socket providing connection to the window_mgr from main CONIT is treated as a file descriptor in UNIX in a manner equivalent to the UNIX primary output (i.e. file descriptor 1), we managed to fool the printf, and putchar functions to make them write to the window_mgr socket rather than to the primary output. To achieve that we make the window_connect function close file descriptor 1 before opening a socket to connect to window_mgr. Since UNIX automatically assigns the least unused file descriptor number, the newly created socket will be assigned the file descriptor 1. As a result, the printf and putchar functions, which are programmed to write to file descriptor 1, does not even know that they are now writing to the window_mgr process instead of writing to the primary output process. In the case where the window_connect function fails to connect to the window_mgr process, it will automatically close the newly created socket and then restore file descriptor 1 to be the primary output so that the user may continue in non-window CONIT mode.

Yet another benefit, of running the window_mgr as a separate process over the Berkeley socket interface using the tcp/ip protocol is that the main CONIT process and the window_mgr process may now run on different machine, even if they are geographically separated. One example is that we may run the main CONIT process on a MIT computer and the window_mgr process on a DTIC computer and have them communicate over internet. The network bandwidth required in this scenario is equivalent to the case when user is telneting to MIT and running CONIT in a non-window mode, sending and receiving character-based *input/output*. *Such bandwidth requirement* is minimal when compared to the alternative implementation of absorbing the window_mgr functions into the main CONIT process such that we have only a single CONIT process. In such case the single CONIT process will be responsible for calling the X window routines directly and trying to display the window on a DTIC computer over internet using the X window protocol which require much more bandwidth.

5.2 Simplification and Conversion Efforts

Simplification of the complexities of the tutorial dialogs was achieved in a number of areas. The basic principle was to assume the user could understand, or at least guess at, the proper direction to take if presented with a pithily stated set of choices (recognizing, at any rate, that a lengthy statement of possibilities and rationales, however well stated and 'instructive', was generally a turn off). Thus, most of the initial menus were drastically shortened. Some additional options were given allowing users to request more detailed explanations, if they so desired.

In addition, a short summary, with examples, of the minimal command set required to perform a search was developed and made available as one of the early options for users so inclined. Previous attempts along these lines tended to include more than was absolutely necessary and did not have examples attached. In some cases menus were shortened by having more menus with fewer choices in each.

In the file selection assistance module we had discovered from our previous experimentation that users when presented with a list of alphabetically tagged file categories were expecting to immediately select from that list instead of, as we forced them, to first choose whether to do that selection or follow some other assistance path. In the new staging we simply follow the users' bent and ask for that selection immediately.

In another stage of file selection assistance, after showing users a list of actual files, we had previously given them the option of writing a memo storing some of these file names along with their comments. Our experiments had demonstrated that this 'poor man's window' was more confusing than helpful. Here, too, we gave in to the majority human proclivities and changed the staging to ask for a selection of file names immediately. In both of these cases, as we have sought to do in general, we allow the user the 'out' of returning to a previous menu – or 'punting' to some other choices – if he wishes *not* to choose from the list presented him.

The conversion of various features of the old Multics CONIT from PL1 to the UNIX/C environment proceeded in a number of areas including document display options; search strategy saving, cataloging, and sharing with other users; and user accounting profiles. While some progress was made on these tasks, we see the desirability of considerably more effort in this area.

5.4 Search Narrowing Selector

We have now completed the preliminary design and partial implementation for a new module that, we hypothesize, will make the selection, generation, and execution of *narrowing* tactics much more automatic and, therefore, simpler for the ordinary user. In this module we would ask the user to select from a list the reason that he deems a document to be non-relevant. This exchange between the system and the user is facilitated by the system already having, through dialog with the user, developed a so-called *Boolean Topic Representation (BTR)* as part of a formalized problem description. (The BTR can be derived automatically from the user-given topic phrase – a more detailed consideration of this aspect of the research is given below in Section 5.4.1.) Reasons on the list include: (1) some conceptual factor in the BTR is not covered (sufficiently) in the document; (2) two factors are not in proper relation to one another; (3) an additional conceptual factor (not in BTR) is needed; and (4) general area of document is wrong. For cases (1), (2), and (3) the system would then elicit from the user the actual factor(s) in question.

In some cases the system is then cognizant of the information necessary to suggest an appropriately modified search strategy. For example, in case (3) the indicated modification would be to search for the new factor and intersect the result with the previous search strategy. In case (1) the system would further elicit from the user if the search term matching that factor was inappropriate for reasons of, for example, stemming (too short stem) or poor word selection more basically; depending on the circumstance, an appropriate modification would then be indicated. In case (1b) – some but insufficient coverage – one indicated modification would be to search only on higher level (e.g., title word) fields. In case (2) one

indicated modification would be to insist on a higher degree of proximity between the search components for the factors in question while an alternate modification would be to transform this into case (3) with the *relationship itself* being considered an additional necessary factor. We discuss below the implementation, testing, and possible extension of this kind of analytic assistance.

The top level menu for this module, and the submenus and operations for one option – option 2, a new factor needed – are illustrated below.

file ns; base menu for narrowing selector; get from: HELP WHY IRRELEVANT abbrev: h y irrel

****

Here are some reasons documents may not be relevant to your topic:

++1) One of your n topic factors

(R1)
R2
...
Rn

is not included in the document at all or in the sense you meant.
++2) Some topic factor other than the n above needs to be added.
++3) All factors are there but at least one is not central to the document.
++4) Some factors are not in proper relation to each other.
++5) General area of document is wrong.
Type number of the reason above which applies for you, or one of the following:
++6) I need more explanation on the 5 reasons.
++7) None of the 5 reasons apply.
++8) More than one reason applies.
++9) I don't want to answer (want to do something else).

******************** NB: The topic factors are taken from the TOP level factors of the topic representation.

****

ns_2 The need-new-factor option.

****

Name the missing factor that needs to be added.

****

ns_2R After user names a [non-null] new factor:

You have named missing factor: *user_named_factor*
Your options now:

**1) Put this new factor in your search strategy.
++2) Consider other irrelevance reasons.
++3) Do something else [help do]

****

For ns_2R(2) or ns_2(null) go back to ns.

NB: NW: want to analyze ns_2 response to see if same as or overlap some rep from current main rep or other previous rep.

for ns_2R(1) go to appropriate stage of HELP MORE FACTORS module assuming search to be modified is one associated with current problem rep.

5.4 Automatic Relevance Ranking

5.4.1 Background

A basic form of testing has been that done by us as system analysts as we continue the development of the experimental CONIT along lines of further conversion of the mainframe PL1 version to UNIX/C and the incorporation of additional features of our expert design. In addition to debugging as such, this form of testing provides a first line of evaluation and, at times, leads to modifications of our original design specifications to provide greater ease of use or functionality. During the course of the previous project period we succeeded in developing the workstation version of CONIT to the point where we could start preliminary testing with real users (other than project staff). The bulk of this effort centered around the use of CONIT by five Professors from India who were attending the MIT Center for Advanced Engineering Studies on a one-semester program sponsored by the United Nations to provide these professors with additional information and skills which they could enhance computer utilization by in the field of public administration. These 5 professors, who had varying levels of expertise with computers but essentially no experience with retrieval systems, served as experimental subjects and each made one or two attempts to use CONIT in its incomplete workstation form to search on topics of their current interest.

Although the version of CONIT used in these experiments was still buggy and incomplete (none of the new functionality described below had yet been implemented), we obtained additional confirmation of the conclusions obtained from our last major set of experiments performed with the mainframe CONIT with the incomplete expert version. As we reported for those earlier experiments ([MARC88a] and [MARC88b]), features already included in the early, incomplete version of expert CONIT show potential for significantly higher levels

of search assistance. However, for this potential to be realized several further developments were required, in particular (1) further debugging and smoothing of the user interface to achieve acceptable levels of ease of use and (2) incorporation of additional retrieval assistance functions.

In terms of pure Information Science research, perhaps our most significant accomplishment in the task has been in the further development of our models of the search process and its evaluation. Our previous sub models that concerned the evaluation of recall by several methodologies ([MARC88b]) have now been fully complemented by a sub model expressing the absolute and relative evaluation of search precision as a function of various formal search structures and match criteria. Our original research had pointed out that relevance ranking in Boolean systems was, indeed, possible *without* resorting to statistical techniques by clearly differentiating searches according to the differences along three dimensions of matching criteria: word exactness (cf. stemming and truncation); field level (title, index term, and abstract); and degree of proximity between words. Our new analyses have run through several stages of development. First we demonstrated that considering two or three levels of match precision on each of the three criteria areas led to 18 levels of relevance (which we reduced to 14 through considerations of certain field/proximity dependencies). This was certainly more than fine enough in terms of gradation to dispel the purely binary myth some have maintained about Boolean-based systems and, perhaps, not too multiplex to permit some version of our scheme of actually performing a series of these search variations to do ranking.

However, part of the analysis of our experiments with the Indian professors indicated that for many user problems it was critical to more finely distinguish match criteria as they applied to *individual* search words and their combinations, as opposed to our original, somewhat naive, thought that we could properly consider such criteria as applying across *all* words (e.g., *all* words searched in just *one* subject field). An analysis of these considerations led to an embarrassment of riches: even for a search of only a few words, literally hundreds of search variations were possible, a fact which goes even further to demolish the binary myth but greatly exacerbates our problem of *how to select* which of many variations to look at.

In the final stage of development (so far in this project) of our precision models we have sought to find means to overcome the embarrassment of riches problem. A major aspect of this effort has been to develop a quantitative model of how precision is affected by selection of one or another search criterion among the many possibilities. In this new model we have established so-called broadening or narrowing factors which quantify how the selection of various search criteria is likely to impact precision. These factors are incorporated into a moderately complex set of formulas - multiplicative and exponential in nature - which yield an 'estimated precision (EP)' for any search based on its purely formal characteristics of Boolean structure and matching criteria.

Leaving aside the question of how valid the EP values are (even on average) in an *absolute* sense, we believe their values in a *relative* sense provide a very useful *a priori* estimate of the *relative* effects of certain variations in search criteria. We have utilized these numerical

estimates as part of our new scheme for selecting only a few of the potentially many search variations as those most useful in ranking the searches (and, therefore, the *documents*) on a given topic along a scale of estimated precision or relevance. In this new scheme we propose to rank automatically for the user the documents in a search according to their estimated relevance in approximately 5 to 10 subsets of decreasing estimated relevance. Elaborations of the current status and plans of our efforts along these lines are given below.

5.4.2 Relevance/Precision Sub Model

In our recent work we have now developed a search sub model complementary to the recall model described above for estimating precision of a search based similarly on purely *a priori* formal features of the search strategy. In our model precision is estimated based on the strength of the matching criteria in 3 dimensions: word exactness, field importance, and proximity. Precision is defined as the fraction of retrieved documents judged relevant to his search topic by some user. Searchers will judge documents as being more or less relevant; i.e., there is a *degree of relevance.* For whatever threshold of relevance used in calculating precision, we demonstrate below that any procedure for *estimating* the precision of a search can also be used to *rank* documents retrieved by *different* searches as to their likely relative relevances. As explained below, our model makes the precision estimate based on certain parameters reflecting the average degree of the broadening or narrowing effect of each match criterion.

The first 2 criteria apply to individual topic (word) searches for which we define parameters as follows:

Word exactness:
- Level E1 (exact match): 0.8
- Level E2 (truncated search on user's FULL term): 0.7
- Level E3 (truncated search on user's STEMMED term): $(0.7)(0.95^{d})$
  - where d = number of characters dropped in stemming operation;
  - nb, for stem = full word, E3 = 0.7 = E2.

Field importance:
- Level F1 (search only title words): 1.0
- Level F2 (search title plus any topic indexes [DE,ID,CF]): 0.8
- Level F3 (search full basic index [all topic fields]): 0.6

The 3rd dimension, proximity, applies to search *combinations* with parameters, as follows:
- Level P1 (strict adjacency [prox W]): 0.6
- Level P2 (same field or sentence [prox F]): 0.8
- Level P3 (any locations [simple AND]): 1.0

These parameters may be thought of as *broadening/narrowing factors [B]*; i.e., the multiplicative fraction by which the estimated precision is changed as a search is broadened or narrowed by particular operations. Thus, the foregoing B values indicate that changes in field level are more critical, on average, to precision than changes in exactness – unless d

is large. In particular, the estimated precision (EP) for a single word search is obtained by multiplying the B values for each match dimension. Take 3 examples:

exact title word search would have EP = 0.8 x 1.0 = 0.8;
an exact word search in full record would have EP = 0.8 x 0.6 = 0.48;
for a truncated stemmed (d=1) search in full record EP = 0.665 x 0.6 = 0.399.

By extension, a n-word search has EP calculated by multiplying the EP values for each word, taking the n^2 root of the result, and modifying the resultant 'base' EP (EPb) by using the B values as follows:

Define the 'imprecision' value (EI): $EI = 1 - EPb$;
get the revised EI (EI') by multiplying by the B value (Bv): $EI' = (EI)(Bv)$;
get the final EP: $EP = 1 - EI'$

(The rationale for the n^2 th root is that while multiplying the n values keeps the *relative* EP values in the right ordering, it tends to reduce the absolute EP which is contrary to our experimental analysis that demonstrates that P increases as coordination level [n = number of term factors] increases. An n th root would preserve the absolute level, so we use the n th root of the n th root [n^2 root = $1/n^2$ power].)

Thus, if

S1 = find ti cat (cat:)	(E3[0], F1; EP=(.7)(1)=.7)
S2 = find dogs (dog:)	(E3[1], F3; EP=(.665)(.6)=.399)
and S3 = S1 AND S2	(P3, B value = 1.0)

then $EPb(S3) = ((EP(S1))(EP(S2)))^{1/4} = ((0.399)(0.7))^{.25} = .72697$

and, since B(P3)=1, EP=EPb.

NB: Levels P2 or P3 of proximity can induce higher levels of field searching than are explicit. E.g., in the above example if a prox w (level P1) instead of AND (P3) were done, the only way a 'cat:' could be adjacent to a 'bird:' in title would be for 'cat:' to be in title also. Therefore, the F level for S1 as a component of S3 is really F1 (not F3).
Thus for S4 = FIND TI cat W birds
EPb(S4)= $((.665)(.7))^{.25}$ = .826
and EP(S4) = 1 - ((1-.826)(.6)) = .8956

A negation combination (AND NOT) EP is calculated as for the intersection search (AND). For a union combination (OR) the EP is taken as the *average* of the component EP's.

5.4.3 Implementation Scheme

As we have indicated above, the final design of an automatic ranking scheme has proved

to be a challenging task. Our current design, which we summarize below, is an attempt to utilize our new model of search precision while limiting the processing requirements to practical dimensions.

For simplicity we assume that the search to be ranked, Sg, is the product of n word factor searches:

Sg = W1 AND W2 AND ... AND Wn

(i.e., it is the default CONIT search strategy for search on user's topic phrase). Now let's define the *primal modification searches (P's)* in four classes - A, B, C, and D - as follows:

n Ai regular topic searches on the n word factors (i.e., the Wi)
n Bi searches on n words in title field only
n-1 Ci pairwise adjacency proximity searches on the Ai
n-1 Di pairwise adjacency proximity searches on the Bi

(NB there are total of 4n-2 P's)

The modified searches (S'g's) are then combinations (Boolean ANDs) of the form:

P1 AND P2 AND ... Pj ... AND Pf

such that

P1 is either D1, C1, B1, or A1;
Pf is either D(n-1), C(n-1), Bn, or An;
and P(j+1) takes on, in turn, all (and just) the following values:
if Pj is Dk, P(j+1) is D(k+2), D(k+1), C(k+2), C(k+1), B(k+2), or A(k+2)
if Pj is Ck, P(j+1) is D(k+2), D(k+1), C(k+2), C(k+1), B(k+2), B(k+1), or A(k+2)
if Pj is Bk, P(j+1) is D(k+1), C(k+1), C(k), B(k+1), or A(k+1);
if Pj is Ak, P(j+1) is D(k+1), C(k+1), B(k+1), or A(k+1).

Note that in 'A1 AND D1' or 'A1 AND C1' A1 is redundant since both D1 and C2 subsume the presence of W1. These, and similar redundancies for the B's, are eliminated by the above rules.

There is, then, a fairly simple algorithm for generating all the S'g's:

First, order the P's:
D(n-1), D(n-2), ... D1, C(n-1), C(n-2) ... C1, Bn, B(n-1) ... B1, An, A(n-1),... A1.

Next, for P1 of Sg1 start with highest allowable P (D1). For P2 take highest allowable

next P (This will be D3 for $n > 3$, D2 for n=3). Continue until we get a legitimate Pf; this completes S′g1.

Now back up to last position where a P was chosen where it is possible to choose a *lower* ranking P and choose it instead. If this completes a legal Pf, we have next S′gi. If not, continue choosing P's until we do. Loop on this paragraph until all S′g's allowable are generated.

For example, for n=4 a few of the S′g's are:

1- D1 AND D3
2- D1 and D2 and D3
3- D1 AND D2 AND C3 AND B4
4- D1 AND D2 AND C3
5- D1 AND D2 AND B4
6- D1 AND D2 and A4
⋮
N- A1 AND A2 AND A3 AND A4 (i.e., the original search, Sg)
where *N* is the number of modified searches

Note that we have limited the number of modification variations, at least at this juncture, through four main *techniques:* (1) *eliminating consideration* of word exactness; (2) limiting the spectrum of field and proximity searching to their extremes (F1 or F3 and P1 or P3, respectively); (3) considering the proximity of only adjacent pairs; and (4) eliminating alternate variations which, in a retrieval sense, are logical equivalents (thus it would be redundant to have a D1 primal [TI W1 adj TI W2] along with a B2 or A2 primal since the D1 must have W2 in title which subsumes the other two primals). Furthermore, another - likely major - reduction is obtained by ignoring any *potential* variation that contains a null primal.

The next step is to calculate the estimated precision of each modified search and sort the searches in EP order, highest to lowest (let us designate this order as S″gi, i = 1,...,N).

The next step is to run the highest EP searches until there are Nd non-null results (Nd is a parameter, tentatively set at 6, sufficient to give an 'interesting' number of EP variations). These non-null results may be designated S‴gi, i = 1,...,Nd.

Finally, the computer generates so-called *differential* (SD) and *cumulative* (SC) ranked searches as follows:

SD1 = SC1 = S‴g1
SD2 = S‴g2 AND NOT SC1; SC2 = SC1 OR S‴g2
SDk = S‴gk AND NOT SC(k-1); SCk = SC(k-1) OR S‴gk
SD(Nd+1) = Sg AND NOT SCNd (i.e., everything else); SD(Nd+1) = Sg

If any SD is null, an additional $S'''g$ would be generated in order to keep the number of ranked levels at Nd+1 (7).

The computer would then be in the position to display for the user the ranking algorithm results in terms of a list of 7 differential searches containing subsets of the original search, Sg, ranked according to their estimated relevance (this relevance figure would simply be taken as the estimated precision of the search which could be shown along with the *numbers* of documents in each subset). A corresponding list would be available giving the cumulative subsets with rankings *down to* a given estimated relevance. It would then be possible, of course, for the user (or the system, in a more automatic mode) to display actual documents from any subset.

6. RECOMMENDATIONS

We believe that the results of this investigation, along with supporting evidence in the fields of Information Science and Computer Science and Technology, clearly indicate that newly emerging advanced techniques in expert retrieval assistance and in computer interface and networking technology put us on the threshold of a new era of vastly more effective and efficient information retrieval capabilities for both experienced and inexperienced, casual searchers. In the retrieval assistance area we anticipate facilities by which information retrieval of bibliographic and text-based files can be advanced from an intuitive art toward a rational, quantified decision-making process. In the interface and networking areas we project the increasing capabilities of graphical user interfaces and rapid, reliable connectivities will enable these expert assistance techniques to be usable at disparate user sites with very friendly, yet powerful, interactive support mechanisms.

However, in order to take advantage of these exciting new potentialities, it will be necessary for some group(s) to pursue a vigorous program of further research, testing, and continued development. The DGIS Gateway program, as developed and supported by DTIC and other government agencies, is in an admirable position to lead the charge toward this new level of capabilities which could be one of the focal points of maintaining U.S. leadership in the information utilization areas which are the basis for societal prominence in the current post-industrial society.

In particular, we recommend that the techniques we have been developing and investigating be quickly incorporated into an 'operational prototype' modality in which form they can be tested on a variety of users. This kind of testing can accrue two types of benefits: (1) widespread experimentation and analysis will promote rapid development of the techniques for achieving the new levels of capabilities we project and (2) even with only limited additional polishing, these new techniques should be exciting enough to the experimental users as to enlist them as a pressure group for wider dissemination, thus leading to the support for funding the overall endeavor.

While the details of a program of the kind we describe above will require some additional effort to elaborate, we have already envisioned some of the next stages of the development of more sophisticated assistance techniques which we outline in the remainder of this section.

6.1 Search Broadening for Recall and its Estimation

We have now developed in more closed form the general design of a scheme for a system-guided method of both (1) selecting the most appropriate search broadening tactics and (2) deriving a very accurate estimation of recall based on user review of particular broadened search samples. This method starts with the recognition that there is a basic asymmetry in the operations of narrowing and broadening in the sense that for the former the user *ipso facto* must *already have* seen (or could easily display from *already retrieved* sets) examples of

the undesired irrelevant documents to be excluded whereas in the latter case the retrieved documents, by themselves, do not necessarily contain explicit clues to the missing documents. There is one universal, and highly effective, tactic for raising recall (discounting, for the moment, any precision-reducing effect): reduce the coordination level by dropping one or more search factors. (This still works in the unusual case where there is only *one* factor – namely, take the *whole database* as the broadened search.)

Specifically, then, assuming there are n search factors (in a Boolean intersection sense) in the candidate search, the system will select samples from each of the n searches obtainable by dropping each of the factors in turn. (We also assume that precision estimates have been made on the candidate search by user relevance judgments of (at least sampled) documents retrieved by the search.) The system then selects *new* documents (i.e., not in the candidate search set) from the samples, presents them to the user for relevance judgments, and calculates (extrapolated) numbers of *additional* relevant documents in the samples as compared with the original search. Employing dialog with the user, the system obtains some additional information with regard to those newly found relevant documents: There are two general cases: (1) the user states that the missing factor is not really essential as a factor in the BTR (in which case the appropriate broadening technique is obvious) or (2) the conceptual factor relating to the dropped search factor is *actually present* in the document.

Case 2 breaks down into: (2a) where the document record has some *new* term associated with the 'dropped' concept not included in the alternate terms previously used to capture that concept – in which case the appropriate broadening tactic is, clearly, to add that as an additional alternate term; and (2b) where the concept *is* represented by *some form* of the terms previously used, but the search formulation was too narrow – in which case the appropriate broadening tactic is to broaden the formulation in the appropriate way (e.g., by lowering proximity, field level, or word exactness specifications).

Before any broadening modification tactic were actually chosen there would need to be a decision taken by the system-user team whether the expected recall increase was worth the expected precision reduction. (Of course, to have reliable estimates of these expectations will require sufficiently large sized samples.) Note that whether or not any search modification is actually performed, the system can now derive another estimate of the total recall base. It is our hypothesis, based on considerable previous analysis (see, e.g., [OVER74]), that this particular methodology for estimating recall can be quite accurate.

6.2 Database Selection from Directory Search

We have recently designed the general outline of a new technique for identifying databases appropriate for searching a given topic. The basic idea is to search in a directory of databases (e.g., DIALOG's) with a topic *field* phrase using the same keyword-stem techniques we have found successful in searching the user's *topic* phrases in the document-containing databases. The user employing this technique would be asked to give a phrase expressive of the field his search topic is in. The explanation to the user at this point would, presumably, give an example or two like "nuclear physics" or "sports medicine." The directory search on

this phrase would then give a set of records for databases of potential relevance. A further elaboration of this technique would be to *rank* the databases with the automatic ranking process to be described below.

6.3 Other Search Enhancements

Ranking can be made more sophisticated (and, hopefully, accurate) in a number of ways. More refined ranking should be possible through explicit consideration of exactness in word matching and intermediate stages in the proximity and field level criteria. Of course, this refinement has to be traded off against the additional cost of processing (and, in particular, in our current schema, the costs of the additional searches required). However, we can readily identify at least two situations in which this greater refinement would immediately be valuable, even if the cost were not insignificant: (1) where the modification space is too sparse (because of null title and adjacency proximity results) to provide the minimum degree of differentiation we posited and (2) where the user explicitly desires (and is willing to pay) to see a greater refinement either *within* the modification space already analyzed or in areas of that space *outside* (e.g., at lower levels of estimated relevance) the previously explored areas. One should note that these notions involve a considerably greater degree of flexibility in the ranking functionality than our initial design which attempts, as efficiently as possible, to perform searches that would identify just (approximately) 6 highly (but decreasingly) relevant subsets of a given search.

The identification of which words *in the user's topic statement* should be considered together as a phrase representing a *single* concept has already been observed in our preliminary analysis to be highly important to the search assistance process. For example, the words 'expert' and 'system' often fall into that category whereas 'smoking' and 'cancer' generally would not. Switching from a simple intersection (Boolean AND) to a word adjacency specification is, in the former case, likely to yield a much higher increase in precision than in the latter case, for which our precision formulas were designed. Also, the loss in recall will be much less when a true phrase is identified.

Beyond the very general explanations CONIT gives the user in identifying phrases as search terms for individual concepts, the current design has just two heuristics for assisting in identifying the most likely combinations: (1) dropping non-significant (function) words and (2) looking at the word pairs found adjacent in the user's natural language topic expression. (If one considers only 2-word phrases from an n-word sequence, the latter selects just n-1 combinations from the n(n-1)/2 total.) A number of additional heuristics worthy of consideration might be explored. First, use the function words in the user's search expression as separators of concepts – e.g., 'networking of information retrieval in office systems' would yield just two word pairs ('information retrieval' and 'office systems') as the most likely candidates. Second, select matching phrases from a thesaurus. Third, consider the adjacency statistics themselves; thus, a retrieval set much larger than one would expect from purely random (independent) assumptions would be a good candidate. Note that these last two heuristics may not reduce the need for generating search sets *per se* but they could still be useful in dynamic modifications to the precision/relevance ranking.

Our recall model does not yet consider the term exactness specifications of the alternate terms for a concept. A more comprehensive analysis should lead to including these considerations as well as to *integrating* the recall and precision sub models into a unified *precision-recall* entity.

Of course, beyond the consideration of purely formal characteristics of the search strategy *per se*, it is advisable to take account of the statistics of the search terms in the database. Thus, for example, one would expect typically that lesser frequency terms would yield more relevant results than higher frequency terms. In this way one can utilize information in the database itself to augment the knowledge base for the expert system.

Also, our estimation models are not well integrated in another sense. The *a priori* recall and precision/relevance models consider only the formal characteristics of the search strategy. The estimations derived from user relevance judgments, on the other hand, do not yet incorporate formal clues. One would desire, for example, to use some of the relevance judgments to dynamically modify the *average* modification parameters according to the *actual* conditions and results of the current topic and searcher. Thus, for example, if the user identifies overstemming as the cause of a precision failure, one would be led to consider increasing the word exactness broadening parameters (E2 and E3) for *this situation.* (Thus, these parameters would get assigned *dynamically.*) Similarly, if a particular broadening tactic – say searching on lower level fields – results in relatively high precision for the additional documents retrieved, the indicated modification would be to *decrease* the values of F2 and/or F3 in our model for the case at hand (as well as make corresponding dynamic modifications to any analogous parameters in an expanded recall model). One can imagine many other factors influencing the parameters such as, for example, a topical, personal, or general *history* of the effect of particular terms on the recall and precision measures.

The latter considerations lead to the further desire to use *accumulated* relevance statistics based on user judgments from *a number of* documents. While we hope to show that modification decisions based on even a *single* document may often be appropriate as well as efficient, there is no doubt that evaluations based on larger statistical samples will be more accurate. We have some preliminary thoughts on extending our existing search structures to include relevance (and irrelevance) judgements on individual documents from the retrieval set.

7. CONCLUSIONS

The objective of this investigation was to further develop and test advanced ret rieval assistance techniques within the framework of the experimental CONIT testbed in preparation for their possible incorporation into the DGIS gateway system. Significant results were achieved in three areas: (1) a partial implementation of an integrated, networkable X window interface that demonstrates many of the advantages of this mode of interaction while preserving the code and functionality of the existing retrieval facilities and allowing users to select the mode of interface that suits their terminal/workstation capabilities and their own proclivities; (2) the implementation of the first phase of an algorithm that automatically ranks documents from a given search according to their estimated relevance to the search topic based on the degree of exactness of match according to our newly refined precision/relevance models; and (3) a detailed set of recommendations leading to the incorporation of existing advanced techniques into operational vehicles, the further testing of techniques, and the design and development of still more powerful new assistance techniques. Two other noteworthy developments were (1) the design and partial implementation of an automatic search strategy narrowing selector based on user feedback or reasons for document irrelevance; and (2) the further simplification of CONIT modules and the further conversion of some modules from the original Multics PL1 code to UNIX/C code.

8. ACKNOWLEDGEMENTS

The author wishes to acknowledge the support of the National Science Foundation whose grant IRI-88-5260 provided the 'seed money' for the initiation of this work. Thanks are also due to the Dialog Retrieval Services for their grant of free computer time for a portion of our investigations. The continued encouragement for our efforts by Gladys Cotter, formerly at DTIC and currently at NASA, was much appreciated. I would also like to thank Mr. Allan D. Kuhn, formerly Director, Hypermedia Laboratory, Defense Applied Information Technology Center (DAITC) (since abolished). Mr. Kuhn saw the the long range ramifications of the integration of artificial intelligence and graphics systems technologies, including the MIT X-Windows development, into an information gateway system. His support of the concepts of CONIT are extensively appreciated. Others associated with the DGIS project at one time or another who have provided important assistance include Duc Tran and Cynthia Shockley.

On my own staff I would like to note particularly Hing Fai Louis Chong who has provided the basis of our X window analysis and development and Scott J. Marcus who performed the bulk of the field searching and relevance ranking programming. Current students who have aided the project include Sabet A. Chowdhury, Sinming Law, and Pui-Man Maria Chong. Former students who chipped in with contributions include Ricardo A. Cardenas, Aleks Gollu, and Man-Wah Colina Yip.

9. REFERENCES

[BATE86] Bates, Marcia. "Variety Enhancement for the Online Subject Searcher." *Abstracts of Papers, Second Conference on Computer Interfaces and Intermediaries for Information Retrieval.* May 28-31, 1986; Boston, MA. Defense Technical Information Center Report DTIC/TR-86/5 2:241-262.

[BATE79] Bates, Marcia J. "Information Search Tactics." *Journal of the American Society for Information Science.* 30(4):205-214. July, 1979.

[BELK86] Belkin, Nicholas J. "What Does it Mean for an Information System to be Intelligent." *Abstracts of Papers, Second Conference on Computer Interfaces and Intermediaries for Information Retrieval.* May 28-31, 1986; Boston, MA. Defense Technical Information Center Report DTIC/TR-86/5 2:12.

[BELL87] Bellardo, Trudi; Saracevic, Tefko. "Online Searching and Search Output: Relationships between Overlap, Relevance, Recall and Precision." *Proceedings of the 50th ASIS Annual Meeting.* 24:11-13; October, 1987.

[BISW85] Biswas, Gautam; Subramanian, V.; and Bezdek, James C. "A Knowledge Based System Approach to Document Retrieval." *Proceedings of the Second Conference on Artificial Intelligence Applications.*" 2:455-460. IEEE Computer Society. Miami Beach, Florida. December, 1985.

[BISW87] Biswas, Gautam; Bezdek, James C; Subramanian, Viswanath; and Marques, Marisol. "Knowledge-Assisted Document Retrieval: I. The Natural Language Interface; II. The Retrieval Process." *Journal of the American Society for Information Science.* 38(2):83-96; 97-110. March, 1987.

[BORG89a] Borgman, Christine L. "The Design and Evaluation of a Front-End User Interface for Energy Researchers." *Journal of the American Society for Information Science.* 40(2):99-109. March, 1989.

[BORG89b] Borgman, Christine L. "All Users of Information Retrieval Systems are Not Created Equal: An Exploration Into Individual Differences." *Information Processing and Management.* 25(3):237-252; 1989.

[BRAJ85] Brajnik, Giorgio; Guida, Giovanni; and Tasso, Carlo. "An Expert Interface for Effective Man-Machine Interaction." in *Cooperative Interactive Systems*, L. Bolc (Ed.), Springer Verlag, 1985.

[BRAU89] Braundet, Marie-France. "Outline of a Knowledge-Base Model for an Intelligent Information Retrieval System." *Information Processing and Management.* 25(1):89-115; 1989.

[BURT89] Burton, Hilary D. "The Livermore Intelligent Gateway: An Integrated Information Processing Environment." *Information Processing and Management.* 25(5):509-514; 1989.

[CHON88] Chong, Hing Fai Louis. "Upgrading a Retrieval Assistance System to a Workstation/Windowing Environment." Master of Science Thesis in Electrical Engineering and Computer Science, Massachusetts Institute of Technology. Cambridge, MA. September, 1988.

[COTT86] Cotter, Gladys A. "The DoD Gateway Information System: Protoytpe Experience." Proceedings of the Seventh National Online Meeting. 7:85-90. New York; May, 1986.

[CROF87a] Croft, W. Bruce. "Approaches to Intelligent Information Retrieval." *Information Processing and Management.* 23(4):249-254; 1987.

[CROF87b] Croft, W. Bruce. "I^3R: A New Approach to the Design of Document Retrieval Systems." *Journal of the American Society for Information Science.* 38(6):389-404; November, 1987.

[DOSZ80] Doszkocs, Tamas; Rapp, Barbara A; and Schoolman, Harold M. "Automated Information Retrieval In Science and Technology." *Science.* 208:25-30; April 4, 1980.

[FIDE86] Fidel, Raya. "Towards Expert Systems for the Selection of Search Keys." *Journal of the American Society for Information Science.* 37(1):37-44; January, 1986.

[FOX86] Fox, Edward. "Development of the CODER System: A Testbed for Artificial Intelligence Methods in Information Retrieval." *Information Processing and Management.* 23(4):341-366; 1987.

[FOX89] Fox, Edward A.; Koll, Matthew B. "Practical Enhanced Boolean Retrieval: Experiences with the SMART and SIRE Systems." *Information Processing and Management.* 24(3):257-267; 1988.

[FREI83] Frei, H. P.; Jauslin, J.-F. "Graphical Presentation of Information and Services: A User-Oriented Interface." *Information Techology: Research and Development.* 2:23-42; 1983.

[GAUC89] Gauch, Susan; Smith, John B. "An Expert System for Searching Full Text." *Information Processing and Management.* 25(3):253-263; 1989.

[HORO80] Horowitz, A.J.; Eastlake, D.E. III; and Low, D. "An Intelligent Terminal for Access to Heterogeneous Chemical Information Systems." *Proceedings of the 43th ASIS Annual Meeting.* 17:83-85 October, 1980.

[KUHN88] Kuhn, Allan D. "DoD Gateway Information System (DGIS) Common Command Language: the Decision for Artificial Intelligence." *Proceedings of RIAO 88: User-Oriented*

Content-Based Text and Image Handling. 863-882. MIT, Cambridge, MA. March 21-24, 1988.

[LEE85] Lee, William Tiger. "Computerized Search Assistance for Document Retrieval." Master of Science Thesis in Electrical Engineering and Computer Science, Massachusetts Institute of Technology. Cambridge, MA. June, 1985.

[MARC81a] Marcus, R.S.; Reintjes, J.F. "A Translating Computer Interface for End-User Operation of Heterogeneous Retrieval Systems; Part I: Design; Part II: Evaluations". *Journal of the American Society for Information Science.* 32(4):286-303; 304-317. July, 1981.

[MARC81b] Marcus, R.S. "An Automated Expert Assistant for Information Retrieval." *Proceedings of the 44th ASIS Annual Meeting.* 18:270-273; October, 1981.

[MARC83a] Marcus, R.S. "Computer Assisted Search Planning and Evaluation." *Proceedings of the 46th ASIS Annual Meeting.* 20:19-21; October, 1983.

[MARC83b] Marcus, R.S. "An Experimental Comparison of the Effectiveness of Computers and Humans as Search Intermediaries." *Journal of the American Society for Information Science.* 34(6):381-404. November, 1983.

[MARC85] Marcus, R.S. "Development and Testing of Expert Systems for Retrieval Assistance." *Proceedings of the 48th ASIS Annual Meeting.* 22:289-292; October, 1985.

[MARC86] Marcus, R.S. "Design Questions in the Development of Expert Systems for Retrieval Assistance." *Proceedings of the 49th ASIS Annual Meeting.* 23:185-189; October, 1986.

[MARC88a] Marcus, R.S. Experimental Evaluation of CONIT in DGIS Environment. MIT Laboratory for Information and Decision Systems Report LIDS-R-1746. February, 1988. (Also Defense Applied Information Technology Center Report DAITC/TR-88/012. Defense Technical Information Center Report DTIC/TR-89/7. NTIS Report AD-A204-721.)

[MARC88b] Marcus, R.S. "Expert Retrieval Assistance Development and Experimentation." *Proceedings of the 51st ASIS Annual Meeting.* 25:115-119; October, 1988.

[MARC89] Marcus, R.S. Book Review: *Knowledge Engineering: Expert Systems and Information Retrieval* by I. Wormell (ed.) in *Information Processing and Management.* 25(4):454-456; 1989.

[MCCU85] McCune, Brian P.; Tong, Richard M.; Dean, Jeffrey S.; and Shapiro, Daniel G. "RUBRIC: A System for Rule-Based Information Retrieval." *IEEE Transactions on Software Engineering.* SE-11(9):939-945. September, 1985.

[MEAD89] Meadow, Charles T; Cerny, Barbara A.; Borgman, Christine L; and Case, Donald O. "Online Access to Knowledge: System Design." *Journal of the American Society for Information Science.* 40(2):86-98. March, 1989.

[METZ89] Metzler, Douglas P; Haas, Stephanie W; Cosic, Cynthia L.; and Wheeler, Leslie H. "Constituent Object Parsing for the Information Retrieval and Similar Text Processing Problems." *Journal of the American Society for Information Science.* 40(6):398-423. November, 1989.

[MISC86] Mischo, William H. "End-User Searching in an Online Catalog Environment." In: Jacobson, Carol E; Witges, Shirley A., comps. *Abstracts of Papers, Second Conference on Computer Interfaces and Intermediaries for Information Retrieval.* May 28-31, 1986; Boston, MA. Defense Technical Information Center Report DTIC/TR-86/5 241-262.

[OVER74] Overhage, Carl F. J.; Reintjes, J. F. "Project Intrex: A General Review." *Information Storage and Retrieval.* 10(5):157-188. 1974.

[POLL81] Pollitt, A.S. "An Expert System as an Online Search Intermediary." *Proceedings of the 1981 International Online Meeting.* 25-32. London. December 7, 1981.

[POLL88] Pollitt, Arthur Steven. "MenUSE for Medicine: End-User Browsing and Searching of MEDLINE via the MeSH Thesaurus." *Proceedings of RIAO 88: User-Oriented Content-Based Text and Image Handling.* 547-573. MIT, Cambridge, MA. March 21-24, 1988.

[RAU88] Rau, Lisa F. "Conceptual Information Extraction and Retrieval from Natural Language Input." *Proceedings of RIAO 88: User-Oriented Content-Based Text and Image Handling.* 424-437. MIT, Cambridge, MA. March 21-24, 1988.

[SALT83] Salton, Gerard; McGill, M.J. **Introduction to Modern Information Retrieval.** McGraw-Hill, New York, NY. 1983.

[SARA89] Saracevic, Tefko; Su, Louise T. "Modeling and Measuring User-Intermediary-Computer Interaction in Online Searching: Design of a Study." *Proceedings of the 52nd ASIS Annual Meeting.* 26:75-80 October, 1989.

[SHOV85] Shoval, Peretz. "Principles, Procedures, and Rules in an Expert System for Information Retrieval." *Information Processing and Management.* 21(6):475-487 1985.

[SMIT86] Smith, Linda. "Machine Intelligence vs. Machine-Aided Intelligence as a Basis for Interface Design." *Second Conference on Computer Interfaces and Intermediaries for Information Retrieval.* 2:13. Boston; May, 1986. (Defense Technical Information Center Report.)

[SMIT87] Smith, Linda. "Artificial Intelligence and Information Retrieval." *Annual Review of Information Science and Technology.* 22:41-77. 1979.

[SMIT89] Smith, Philip J; Shute, Steven J.; Galdes, Deb; and Chignell, Mark H. "In Search of Knowledge-Based Search Tactics." *Proceedings of the Twelfth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval.* 3-10. June, 1989.

[SULL90] Sullivan, Michael V.; Borgman, Christine L.; and Wippern, Dorothy. "End-Users, Mediated Searches, and Front-End Assistance Programs on Dialog: A Comparison of Learning, Performance, and Satisfaction." *Journal of the American Society for Information Science.* 41(1):27-42. January, 1990.

[WEIN86] Weinberger, Marvin. "Knowledge Gateways: A State-of-the-Art in Information Integration." *Beyond Databases: Knowledge Delivery Systems Come of Age.* Industry Industry Association Seminar. Cambridge, MA. August, 1986.

[WILL83] Williams, Phillip W. "A Microcomputer System to Improve the Recall Performance of Skilled Searches." Proceedings of the 1983 National Online Meeting. 4:581-590. New York; May, 1983.

[WILL86] Williams, Martha E. "Transparent Information Systems Through Gateways, Front Ends, Intermediaries, and Interfaces." *Journal of the American Society for Information Science.* 37(4):204-214; July, 1986.

[WOOD86] Woodsmall, Rose Marie; Anderson, John.E. "NLM Introduces GRATEFUL MED: A User-assisted Interface to MEDLINE and CATLINE." *The NLM Technical Bulletin.* No. 202:1-8; February, 1986.

[YIP81] Yip, Man-Kam Kenneth. An Expert System for Document Retrieval. Master of Science Thesis in Electrical Engineering and Computer Science, Massachusetts Institute of Technology. Cambridge, MA. February, 1981.